Copy #44

PRELIMINARY ENGINEERING DESCRIPTION

"STORED PROGRAM ACCOUNTING AND CALCULATING EQUIPMENT" (SPACE)

June 5, 1958

MEMORANDUM TO: Mr. F. J. Furman

SUBJECT: SPACE Preliminary Engineering Description

Attached is your copy of the preliminary engineering description of the "Stored Program and Calculating Equipment" (SPACE).

This description refers to the logic employed in the SPACE accounting machine in its basic form and does not include any of the facilities for expansion such as integrated calculation, increased storage, etc. It should be noted that the basic logic described can be expanded and/or rearranged to provide other machine configurations with particular emphasis upon a universal print edit control unit, a processing unit for an improved RAMAC system, or a calculator in the basic 650 class.

It is important to remember that the attached write up describes the machine in its present status and that efforts to reduce cost and increase reliability may cause changes to be incorporated. However, from the results of various programming studies, it is concluded that the basic philosophy and logical approach is superior to any presently known for the type of machine desired.

This description will be updated as developments in the program warrant.

C E Branscomb

C. E. Branscomb, Manager
Accounting Machine Development

CEB:mk
Attachment

PRELIMINARY ENGINEERING DESCRIPTION

"STORED PROGRAM ACCOUNTING AND CALCULATING EQUIPMENT" (SPACE)

PREPARED BY

ACCOUNTING MACHINE, ADVANCED DEVELOPMENT
DEPARTMENT 191
DEPARTMENT 280
DEPARTMENT 268
ENDICOTT

JUNE 5, 1958

TABLE OF CONTENTS

June 5, 1958

"STORED PROGRAM ACCOUNTING AND CALCULATING EQUIPMENT" (SPACE)

PRELIMINARY ENGINEERING DESCRIPTION

The SPACE logic has been designed to provide a basis for the next generation of accounting machines above the 407 and also, through rearrangement and/or expansion, this system can be considered for other applications including a Print Edit Control Unit for off-line Data Processing Systems, a small calculator in the 604, 607, 608 class, a processing unit for an improved RAMAC system, and an expanded calculator in 650 class.

Essentially the SPACE approach is based on the WWAM logic with the major difference being the incorporation of completely stored rather than control panel programming techniques. The use of stored program techniques offers the advantages of increased programming flexibility, ease of expansion, and, particularly, in transistor circuitry, substantial cost reductions and increased reliability by the elimination of circuits to drive through a control panel and of the control panel contacts themselves.

In its basic form the SPACE machine is functionally equivalent to the 407 but operates at substantially increased speeds. Expansion to optionally include integrated calculating ability can be easily accomplished; however, multiplication by a sub-routine can be performed on the basic machine without any additional circuitry.

Other major optional features will include:

1. Expansion of memory - The total amount and increments of this expansion are as yet not resolved; however, for the accounting machine either 1,400, 2,000, or 4,000 alphanumeric characters probably will be made available.

2. Attachment of a RAMAC File - This feature is being given early consideration.

3. Additional card equipment for input-output - Reader-punch, dual hopper feed, multiple readers, etc.

4. Other input units - Magnetic tape, paper tape, and possibly character sensing.

The following description applies to the machine in its basic form which contains three units: 1) processing unit, 2) reader-summary punch, and 3) printer. Each component is on a separate base; however, all functions are controlled through the process unit to which the input-output equipment is cable connected. Run circuitry for the reader-summary punch and printer will be incorporated in these individual units.

SYSTEM CHARACTERISTICS

A. Reader-Summary Punch

This unit is based on the 088 Collator format with the secondary feed replaced with a 250 cpm punch. The primary feed, including the File Feed, and the five selective radial stackers will be retained.

The configuration of the Reader-Summary Punch is:

HbPR Sel S_4 S_5 S_3 S_2 S_1 Sel R R H

with the reader portion operating on a 20-cycle point basis at either 400 or 800 cpm. This unit will include a two-tooth clutch operating at one-half speed and will require a clutch decision time of approximately 11 ms. The normal reader speed is 800 cpm for a tabulate operation; however, by clutching every other cycle, 400 cpm for listing can be maintained. Extending the calculate time can be accomplished by keeping the reader latched for additional cycles. Since the control of the reader is determined by the program, any or all of these functions can be combined for a single pass of the cards and thus the optimum number of cards processed can be combined for a single pass of the cards and thus the optimum number of cards processed can be achieved.

The punch portion operates at 250 cpm on a 16 cycle point basis. A four-tooth clutch is included on this unit with a clutch decision time of approximately 11 ms.

Stackers 1, 2, and 3 are assigned to the reader and stackers 3, 4, and 5 assigned to the punch. The common stacker, 3, is being considered for merging of cards from both feeds while the others will be used for selection within each respective feed. The detailed logic and circuit design for providing this merge option has not been resolved as yet. Because of the speed differentials between the Reader (800 cpm) and the Punch (250 cpm), timing difficulties for performing this merge operation are evident.

B. Printer

The mechanical portion of the Chain Printer will be utilized as the printer output mechanism operating at 400 lpm on a listing operation. 600 lpm can be achieved by continuous printer operation without interspersed reading. Included in this unit will be the hammers and hammer magnets, a chain location sensing mechanism, a hammer actuation sensing mechanism for checking, and a single speed hydraulic carriage. Location of the hammer drive will be dependent upon the type of circuitry used. If suitable transistors are available to drive the magnets directly, these will be located in the processing unit.

It should be noted that the printer unit will be identical to that being proposed as the mechanical section for other programs; however, these other applications for the printer will require a separate process unit, whereas the SPACE system is so designed that the output circuitry is integrated into the processing unit logic.

A single-speed hydraulic carriage has been specified. It is intended that the two speed drive unit now being developed for the chain printer be included; however, the controls for operating at the higher speed will not be utilized on the basic machine. Optionally these controls and their allied circuitry will be made available.

C. Speeds and Machine Cycles (Chart 1)

The basic machine cycle has been established to provide 400 lpm output on a listing operation and is, therefore, 150 ms. The card reader operating at 800 cpm is capable of clutching every 75 ms. Of the 150 ms basic cycle, 45 ms are required for card reading (12/20 x 75), 80 ms are required for printing, and approximately 4 ms are required for decisions and speed variations so that approximately 21 ms remain for computing. Summary Punching will take place immediately following the print cycle (determined by the clutch latch location) and will require approximately 180 ms. Since this machine is completely unbuffered and continual scanning of memory for printing is necessary, no other function can be performed during print time. During calculate time, only carriage operations may be interspersed. Reading and summary punching may be performed simultaneously by interleaving read and punch scanning of memory.

On a tabulate operation the reader will run continuously with card reading taking place every 75 ms. This provides approximately 29 ms for compute; however, if a decision has to be made whether or not reading should continue, then only 11 ms of compute time are available prior to the making of this decision.

To extend the calculate time beyond the times indicated above, increments of 75 ms (card read cycle) are required. With the completely asynchronous printer, a print cycle may be called for at any time except during card reading.

The carriage requires approximately 19 ms for a single line space and approximately 6 ms for each additional space. This means that the decision to perform a carriage operation must be made early in the compute cycle so as not to hold up printing.

Although not completely resolved at this time, it appears that carriage functions may be programmed for either "before printing" or "after printing" operation depending upon the routine. By operating a card off (printing information from card 1 after reading card 2) approximately 70 ms (time between print cycles) on a list cycle will be available for "before printing" carriage operation. Paper movement times beyond these will require an operation corresponding to that for extended calculation.

D. Processing Unit

All of the machine logic is performed within the process unit containing a core memory consisting of 1400 8-bit characters. The eight bits consist of six bits for the alphanumeric binary code, a word mark bit for defining word length, and the redundant bit for parity checking. This memory stores both the program instructions and the operational data and employs a variable word length concept, each position being addressable. Three distinct areas of memory have been set aside for reading, printing and punching utilizing 280 characters with the remaining 1120 characters available as general and instruction storage.

Data flow through the machine is on a serial-by-digit, parallel-by-bit basis.

The basic memory cycle of the machine is 12 usec. (read to read), this being broken up into six 2 usec. pulses. 8 usec. are set aside from the beginning of read to the end of write with approximately 4 usec. between these available to perform logic. Since the type of memory drive has, as yet, not been resolved, the exact timing requirements have not been established.

Memory

As stated previously, the memory of the basic machine consists of 1400 alphanumeric characters, each containing eight bits. Included above the 80 character positions set aside for the card image input area are two additional 80 core

planes which are read into directly from the reader brushes. These are referred to as the row bit memory planes. A ninth plane of 80 and 120 cores respectively is also included above both the 80 character punch output area and above the 120 character printer output area. Both planes are used for checking the output information and are energized from their respective units.

The memory is decimally addressed from three 3-digit binary-coded decimal registers. Each address register is made up of CTRL latches; 4 such latches for each of the "UNITS" and "TENS" positions and 6 latches for the "HUNDREDS" positions. To advance these registers, an address adder is used.

This adder is made in two sections, one for the low order digit where modification by 1, 2, or 3 (depending upon the operation) is provided, and the other section for modifying the higher orders by one should a carry result from the low order modification. The result of each digit modification is held in individual intermediate registers.

On a memory cycle where a specific address register is being used, the present data in this register is read through the adder and the result is placed in the intermediate register. Following write time of this cycle the data from the intermediate registers is transferred to the address registers. Low order digits are transferred unconditional; however, transfer of the high order digits are under control of the carry gating.

Starting points of each address register, which is the address in the instruction, is read into the register directly from memory. Only one register can be modified during any one memory cycle.

These three address registers are the "I" or instruction register, the "A" register and the "B" register. Although three digits would restrict the addressing to 1,000 positions, the addition of two zone bits to the high order digit permits each register to address 4,000 positions of storage. All three registers are capable of addressing each storage position.

Normally the "I" register scans data from high order to low order while the "A" and "B" counters scan data from low order to high order. Since the adder is capable of either increasing or decreasing the amounts in the registers, all registers can be made bi-directional.

Data Flow (Chart II - Data Flow Diagram)

The 8 memory sense amplifiers are connected directly to an 8 position (7 bits plus WM) register called the "B" register which is made up of 8 current switching latches. All data read out of storage is set-up, serially by digit, in this register at read time and from here either regenerated or switched to other processing functions.

A second single character register is also available called the "A" register. This register also consists of 8 current switching latches which will accept the six binary bits and two C bits, one associated with the zone information and one associated with the digit information. The word mark is not carried through the "A" register.

The outputs from both of these registers are available to the Compare, Print Edit, Sign Control, and Adder sections of the logic and also to the inhibit mix which provides the set-up pulses for the inhibit drivers of memory. Eight inhibit lines (7 bits plus WM) are available from the output of the mix.

Basically, the programming logic employs a two address concept, thus two memory positions can be specified for one program step if required. The first or "A" address usually defines the "from" field and the second or "B" address the "to" field. One memory cycle (12 usec.) is required to process a character from each addressed location; these memory cycles being referred to as the "A" and "B" cycles respectively. A single address instruction will require only one memory cycle.

A two-address program step is accomplished by the following routine: At read time of the "A" cycle, the character at the "A" storage address is read from memory into the "B" register and, at write time of the same cycle, is regenerated into the same storage location. This data is transferred from the "B" register to the "A" register at the end of this cycle.

On the "B" cycle, during read time, the data at the "B" address is read into the "B" register. Between read and write times of this cycle, logical rearrangement (compare, print edit, or add) of the data will take place utilizing the outputs of either or both of the "A" and "B" registers. At write time, the output of either the "A" register, the "B" register or one of the logical sections will be gated through the inhibit mix and written into storage at the"B" address.

The registers are automatically reset prior to the receipt of new information.

Word Mark

Since the machine is based on the variable instruction and variable word length concept, word marks, which are denoted in each case by the presence of a bit in the eighth memory plane in the high order character of a word, are used to define the length of that word. These word marks are set during the program loading operation but may be altered during the routine by programmed instructions. Word marks are read from storage into the "B" register along with the character data and regenerated unconditionally. They are not transferred with the character data. The parity check is so designed that the word mark is included as a character bit; however, since the WM is not transferred, provision is made to adjust the check bit to correspond only to the data available.

Sensing of the word mark bit in the "B" register sets the word mark latch associated with the cycle control functions. When operating upon functional data, the setting of this WM latch will stop the advance of the "A" and "B" counters at the end of the memory cycle during which the WM was sensed and also will prepare the machine to read the next instruction.

On all operations except "Add", the field length is determined by the first word mark sensed. Since an "Add" operation has the format of $\not{\epsilon}A + B = B'$, where the result is written at the "B" address, the field length is specified by the word mark associated with the "B" field. If the "A" field is shorter than the "B" field, the sensing of the "A" word mark will stop further readout of that field; however, so that carries may be propagated, the "add" operation will continue until the "B" field WM is sensed.

When an instruction, under control of the "I" address register, is being read from storage, its length is dictated by the word mark of the next succeeding instruction. The "I" counter advances until a word mark is sensed in the "B" register and the WM latch is set. However, since this data is not associated with the program step under consideration, the character read is regenerated, and the operational portion of the cycle is initiated. The instruction counter remains set at the position where the word mark was sensed, this being the first character of the next instruction in sequence.

Cycle Control (Chart III)

The instruction format is OP AAA BBB; where "OP" is a one character operation code, followed by the three digit "A" address and then the three digit "B" address. This format can be altered to include only the operation code, the

operation code and one address, or the addition of a single test character or digit to any of the above configurations. The test digit is used primarily for branching decisions where it is compared against data from memory specified by the "B" address, for carriage control, or for alteration switch selection.

Since the instruction need only contain the data required to complete a single program step, the variable instruction length concept is evident; however, it is possible to combine more than one operation in one program step if desired. For example, input-output codes require only the operation be specified, therefore, modifications to the routine can be included within this single step. This is accomplished by specifying the modifying address in the "A" field location.

Normally the program is sequential; however, when branching is desired the "A" address specifies the address of the next instruction. When including the test feature within the instruction, the results of the comparison will determine whether the next instruction should be that specified at the "A" address or should be the next in sequence. The "B" address can be used to specify the storage position of comparison data or can be omitted if not required.

Control of the machine cycles during instruction and data processing is accomplished by the Cycle Control which includes the "A"-"B" trigger, the memory timer ring, and the "Instruction-Execute" Trigger. The latter consists of a single trigger circuit ("I-Ex") which is controlled by word marks and determines whether an instruction or data is being processed.

The memory timer ring, which emits subcycle clock pulses within the memory cycle, is used to provide the timed signals to the Cycle Control. For example, the pulse at t_4 time will permit gating of the sense amplifiers so that reading from memory can be accomplished.

Cycle Control combines the operations of the "I-EX" Trigger, the "A"-"B" cycles Trigger, and the memory timer ring so as to provide gating pulses to the various machine functions through the Operations Decode matrix.

Another ring, the eight position "I" Timer, is active during instruction cycles and provides gate pulses for routing instruction information to the Operations Register and the Address Registers. This is a stepping ring and operates at a 12 usec. rate.

All machine functions must be initiated through the reading of an instruction from memory. At this time the "I-Ex" trigger is set on "I" and the "I" address register is activated to read the high order position of the instruction which is the operation code. This data is read into the "B" register from which it is transferred to the operation register latches and also regenerated back into memory. On the next memory cycle the "I" register advances by one and addresses the next higher position of storage which is the high order digit of the "A" word address.

As the "I" address register continues to count up, the "A" address is read and transferred digit by digit from the "B" register to both the "A" address register and the "A" data register and regenerated. If a "B" address is specified, a similar routine is followed except that the "B" address register is now set up. Sensing of a word mark will stop the "I" address register operation and transfer the "I-Ex" trigger to "Execute" thus activating the "A" address register on the next cycle and the Data Flow routine will be followed.

Also, the sensing of the word mark in the "B" character register will prevent the transfer of the contents of this register to the "A" character register, so that it can be seen that the "A" character register will always contain the last character of an instruction. This is necessary, because this last character is a "test digit" for a "Test" instruction, or is a carriage control digit for a "Carriage instruction."

Since the WM will now cause the "I-Ex" trigger to set up the execute portion of the cycle, the data in the "A" register is either transferred to one of the carriage registers or compared with data from memory as specified by the "B" address. If a comparison test, a branching operation will be effected by permitting the "I" address register to remain at the next sequential address or by causing this "I" address to be replaced by the address stored in "A" address register depending upon the result of the comparison.

Since all address registers remain at the next position in sequence from that last processed until another instruction is read, it is possible to perform such operations as "multiple field add" by merely respecifying the operation code on succeeding program steps. For example, an instruction for performing "add" operations on three fields which are written successively in storage would be specified as OP AAA BBB OP OP where the "A" and "B" address are the addresses of the low order digit of the rightmost field to be processed. The operation in all three locations would be "add". Word marks required for cycle control would be placed above each "OP" code.

Operations Register and Operations Decode

So that the operation to be performed may be retained throughout the entire program step under consideration, a single character Operations Register is included. The output of this register is fed to a matrix called Operation Decode where the signals available set up gate pulses to the Cycle Control. Through Cycle Control and timing pulses, the switching of the data paths through the logical functions is accomplished for the operation specified.

Compare

The compare circuit has its input gated from both the "A" and "B" register output lines and is operative on "B" cycles only. Output from the comparison matrix is set into a single latch register and held until immediately prior to read time on the "B" cycle of another compare operation. The output of this latch is returned to the Cycle Control and is used to condition instruction branching for subroutines. Retention of the comparison results is necessary primarily on total control breaks for branching between subroutines on different classes of totals.

Print Edit

The Print Edit function is used to prepare information to be printed in a desired format. This involves zero suppression, floating dollar sign, protection asterisks, sign and total class identification, punctuation, and blank insertion and is performed on a word basis.

With the stored program concept, this function is implemented with the use of an "edit control word" which is inserted into the machine during the loading operation. This control word contains standard (including 8-4, 8-3, and 8-2) characters and can be written so as to provide several formats. For example, one such word might be ###, ###.@@ b CR&*** where "#" indicates zero suppression preceding significant digits; @ indicates a character position which will either be replaced by significant digits or zeros irrespective of preceding significant digits; "," indicates a comma printing of which is conditioned by the number of significant digits; "." indicates an unconditional period; "&" is a test symbol for conditional minus sign or CR printing; "b" indicates a blank insertion; "CR" indicates a credit symbol under control of the sign; and "*" indicates the class of total.

When 010345^{-} (minus sign above the units position) is to be printed on a minor total, it would appear as 103.45 CR *. However, 00678901 on a major total would be edited 6,789.01 ***, the same control word being used in both cases with the exception that when transferring the control word for the minor

total, the leftmost asterisk is addressed while on the major total, the rightmost asterisk is addressed.

Two program steps are required to perform the edit function. The first, which is a transfer operation is used to write the control word (or portion thereof) into the area of print storage that the edited data will appear. The second program step, which is "Print Edit," will contain the address of the low order position of the data to be edited in the "A" field and address of the low order position of the print area, that which contains the control word, in the "B" field.

During the "A" cycle the "to be edited" data is read from storage into the "B" register, regenerated, and transferred to the "A" register. On the "B" cycle, the control word, which is read into the "B" register, and the functional data in the "A" register are compared. According to a fixed set of rules, the edit unit will gate either the "B" register or "A" register output or present a special non-standard character to the inhibit mix for writing into the print storage area occupied by the control character.

This operation continues digit by digit, scanning from low to high order until a word mark is sensed in either field. As this scan is taking place "*", ".", ",", and "CR" symbols in the "B" register take precedence over data so that the "B" register output is gated to the inhibit mix and the "A" address register is not advanced. Thus the low order character of the "A" field is again compared on the next "A-B" cycle with the second character of the control word. This operation will continue until either an "@" or a "#" is sensed in the "B" register. If an "@" is sensed the output of the "A" register is gated to the inhibit mix and both address registers are advanced. Recognition of a "#" symbol will gate the output of the "A" register to the inhibit mix if it is not a zero. If it is a zero, a special non-standard character will be generated by the print edit unit and presented to the inhibit mix. Under this condition both address registers advance for the next "A-B" cycle.

When a word mark is sensed, a reverse scan (high to low order) is taken with the non-standard special characters replaced by the edit unit with blanks until a significant digit is recognized. After this the non-standard special characters are replaced by zeros. The print area for this word is now edited and can be read out to the printer on the next "Print" operation.

Floating dollar and asterisk fill are optional features, and require an additional program step. That is, to float a dollar sign, the data must first be edited (zero-suppressed), then an instruction may be given to float a dollar sign (similarly with "asterisk fill").

These operations follow the above routine except that two more non-standard special characters are used, one for floating dollar and one for an asterisk. For these operations, on the forward scan the first zero encountered after each significant digit is replaced by one of the non-standard characters, depending upon the operation. (In the case of asterisks, all zeros are replaced). On the reverse scan blanks are again inserted until one of the non-standard characters is encountered. At this time the standard symbols replace the non-standard character. As the scan continues, following a significant digit, the non-standard characters are replaced by zeros.

Adder

The adder to be included in the machine will be of the core matrix type. Data, which through the machine is binary coded, is encoded decimally at the input to the adder and decoded back to binary at the adder output. Thus, the adder core matrix is a 10 x 10 array using decimal input and producing a two out of five output. A second 2 x 10 core plane is added to the array for complementing on a subtract operation.

For true add, ($\not A \not + B = B'$ or $-A-B = B'$) on the first "A" cycle the data is read from storage to the "B" register, regenerated, and transferred to the "A" register through the adder by adding zero to it. The "B" word is read from storage on the "B" cycle and the outputs of both registers are then presented to the adder. At write time of this cycle, the adder output is gated to the inhibit mix and written into the "B" address location. A carry effected at this time will be held in a carry latch. On the next "A" cycle, transfer to the "A" register from the "B" register will, by being routed through the adder, be modified in accordance with the output of the carry latch (either added to 1 or 0 depending upon the previous operation). The next "B" cycle will again provide the sum of the "A" and "B" register outputs with the carry propagated as before.

This operation will continue until the word mark in the "A" field is sensed at which time the "A" address register will be stopped and read out of data from that field inhibited. The "B" address register will continue to advance until a no carry is sensed.

Subtraction ($-A \not + B = B'$) is accomplished in a similar manner except that on the first "A" cycle, data is switched through the complement plane and the "10's" complement of the digit in the "A" field is transferred into the "A" register. On the "B" cycle, a true add is performed and the carry again held in the carry latch. The next "A" cycle will again switch data through the complement plane, however,

if a carry was effected, the "10's" complement output will be sensed and transferred to the "A" register. If no carry was effected, the "9's" complement is used. At the end of the subtract program step (WM sensed in "B" field) a test of the carry latch is made and if a carry is effected, a true figure (B > A) is in storage and the operation is concluded. However, if the carry latch does not indicate a carry, then a complement figure (A > B) is in storage and recomplementing is required.

To recomplement, successive "A" cycles are taken of the entire "B" field with the "B" data being read from storage through the complement plane to the inhibit mix for rewriting into storage. During this time, on the first cycle the "10's" complement is forced. On recomplementing, the sign (B zone bit) is reversed.

Checking

Three parity check circuits are provided in the process unit. These are located in such a manner that any single error will be detected following a logical manipulation of data.

Storage readout is checked at the output of the "B" register, which is essentially one input to the other logical functions. A second check is provided at the "A" register output this verifying the "B" to "A" register transfer and also the other input to the logical functions. Since the outputs of all logical functions are mixed in the inhibit mix circuit which is immediately prior to writing in storage, a third parity check circuit is available at that location.

As both a customer and customer engineering aid, single bit error registers are associated with each check circuit. These are manually reset.

Detection of an error will cause the machine to stop at the end of the memory cycle on which the error occurs except during an input or output operation. In these cases the machine will continue to the end of the cycle of the unit in operation.

Separate and different check circuits are provided with the peripheral equipment. These are discussed with the description of the input-output logic.

Card Input

The card reader is a parallel sensing device and contains two sets of 80 column brushes. Each brush is connected directly to a multiple winding core which is set by a hole in the card at each digit time. These cores are contained in the ninth and

tenth planes of storage, called the row bit memory planes, above the card input area; the ninth plane being associated with the second read brushes and the tenth plane with the first read brushes. Data obtained from the tenth core plane (first read) is used for checking only. Functional data to be recorded in the input area of storage is encoded from the ninth row bit plane.

To transfer the parallel row bit recording into serial data for writing into memory all eighty row bit cores in the ninth plane are scanned in 12 usec. intervals after each card digit time. Through a series of binary timed circuit breakers and the coincidence of the row bit pulses, encoding of the Hollerith to binary codes is accomplished. To provide for encode checking, a "C" circuit breaker emits a pulse at digit times which, when encoded, will result in an even number of bits. At nine digit time an unconditional "C" is placed in all card input memory positions so that the blank column parity check will be satisfied. This check bit is modified on succeeding scans as required.

Certain input rules have been established and these are satisfied by logically mixing the row bit pulse data and data already in input storage for that card at each digit time. These input rules are:

1. It is not possible to read in any bit that has already been read into input storage per column.

2. For multiple punched columns, only the highest order digit will be recognized except when this digit is an eight. Recognition of an eight as the highest order digit will also allow the next highest digit only to be recorded.

3. For a laced column in which eight is the highest order digit, only the eight punch will be recognized.

4. A zero zone punch without numeric information will be encoded as a numeric zero (8-2).

5. Only one zone punch in a column will be recognized.

A card feed cycle is initiated by an operation code calling for feed. The card is fed "9" edge first and as it passes under the first set of brushes, the tenth plane row bit cores are set at each digit time in accordance with the holes in the card. Between digit times this row bit plane is scanned and reset and the number of holes sensed is added in a "Modulo 16" counter.

Between card cycles the 15's complement of the resultant sum is read into a second counter and held until the card is read at the second brushes. This is done so that the first counter will be free to accept information from the next card. As the card passes the second brushes, the 9th plane row bit cores are set at each digit time in accordance with the sensed information. This plane is also scanned between digit times and two actions are set up.

The first is the checking operation whereby this row bit count is added to the complement counter holding the data from the first reading. This counter should contain "15" (all triggers ON) at the end of the card cycle or an error in reading is indicated. The second action is that of transferring the card information into input storage. The "A" Address Register is used to address the card input area, and since the row bit planes are immediately above this area, only the one address counter is required. As the row bit plane is scanned, the data in the corresponding position of storage is read out into the "B" register. The output of the "B" register is available at the encoder where the encode C. B. pulses are mixed with the row bit pulses. Logically the following functions are effected to implement the input rules stated previously.

1. An "8" and "1" or a "4" or a "2" in input memory will inhibit recognition of additional digit information.

2. An "A" bit recognized without numeric data from memory and without a "B" bit from the encoder will be changed to an "8" - "2". If both conditions are not satisfied the "A" bit will be recorded.

3. The first zone punch recognized will inhibit recording of further zone data.

4. With an "8" and "4" in memory, the reading of a 1 punch will delete all numeric bits except the "8".

Checking of the encoded and mixed information is provided through the parity check circuit at the inhibit mix. Since data is written into memory on every digit cycle, the parity check is operative 12 times per column during the card read cycle.

Punch Output

The configuration of the punch portion of the Reader-Summary Punch is Hb P R Sel SSS. This unit operates at 250 cpm on a 16 cycle point basis and is mechanically identical to punch units scheduled for production for other programs. The blank station is the normal "first" brush station with the sensing mechanism replaced with suitable card guides. Inclusion of brushes and a contact roll at this location will provide a Reader-Punch unit which will be available as an optional feature.

Since the punch and sensing mechanisms are parallel and output is in the Hollerith code, it is necessary to convert the serial binary coded information from the processing unit. For this, a punch matrix and decoder are used. On a signal to punch, the "A" address register is set to address the rightmost position of punch storage (180) and at each digit time scans all 80 punch storage locations in synchronism with the punch magnets at 12 usec. intervals. The information is read through the "B" register to the punch decode function where, in coincidence with the signals from the punch C. B's, it is converted to the Hollerith code. For example, when the card digit "12" time circuit breaker is made, each punch storage location containing an "A" and a "B" bit will, through the punch matrix, cause the corresponding punch magnet to become energized. This operation continues through each card digit time.

Checking of the punched information is accomplished in a manner similar to that used for the reader. As a coincident signal of the "B" register and the punch C. B's is available at the punch decode output, it is read into a "Modulo 16" counter at the same time it is read into the punch matrix. Through this, the total number of punches which should be made on a "Modulo 16" basis, is recorded in the counter. At the end of the card cycle, the 15's complement of the resultant sum is read into a second counter so that the first counter will be free to accept information from the next card. As the card passes the punch check brushes, the ninth plane of cores above the punch output section are set at each digit time in accordance with the sensed information. This plane is scanned between digit times and the count of these cores is added to the complement counter holding the data from the first reading. This counter should contain "15" (all triggers ON) at the end of the card cycle or an error in punching is indicated.

Print Output

Printed output will be provided by a Chain Printer. The logical controls and hammer drive signals to this unit will be included in the SPACE Processing Unit with the same method of hammer selection being used as proposed for the Chain Printer Control Unit.

Mechanically the printer includes a chain on which the type slugs are arranged in five alphanumeric arrays of 48 characters each. The chain operates at a constant speed without stopping and, therefore, the printing can be termed "on the fly". The hammer action strikes the paper from the rear forcing it against the type.

Print spacing along the printing line is 10 to the inch with a hammer available for each printing position (120 hammers required). The chain pitch is approximately .15 inches between centers. Since the type moves with reference to the hammer positions, the timing of activating an individual hammer is dependent upon the arrival of the required type character at a printing position. The hammers are not activated simultaneously or in a set order, but rather individually as the required type passes the desired print position.

One possible position of the chain and its type with respect to the printing positions is: Type A, C, E, G, I, etc., are in line with the 1, 4, 7, 10, 13, etc., printing positions. If the first position of the memory has an "A" in it, the first position hammer would be activated; likewise, with the 4th position for a "C", etc.

As the chain moves, a "B" will come into the 2nd position. Now the 2nd, 5th, 8th, 11th, etc., positions of memory are investigated for B, D, F, H, etc. Following this, the 3, 6, 9, 12, etc., positions are investigated for C, E, G, I, etc.

At the chain speed of approximately 90 inches per second, the chain moves a distance of 0.00108 inches (when "A" is in position one, "C" is 0.00108 inches from position 4) in 12 usec. This has been determined so as to equal the memory access time of 12 usec. Looking at every third position of memory (40 positions) takes 480 microseconds. This is referred to as a sub-scan. The spacing of the type is so arranged that the second character will reach the second print position 555 usec. after the first character was in the first print position. This allows 75 usec. before the second memory scan must begin to provide for speed variations, resets, etc.

In three print sub-scans, every position of the print storage area will have been interrogated, and each printing position on the paper will have been presented a type face. This is referred to as a Print Scan. In 48 print scans, every character will have been presented to a given print position. Thus, 555 usec. per sub-scan, 3 sub-scans per print scan, and 48 print scans per line of print requires the 80 ms of continual scanning of memory to produce one line of print.

Data flow through the process unit for a print operation is initiated by an operation code "Print". At this time the "B" address register is set to the high order position of the print storage area (201). The character at this location is read through the "B" register, regenerated, and presented to the compare circuit. The Print Character Generator (character counter) will also present to the compare circuit the code for the character which is on the type face at the first printing position. If there is an equal comparison, a pulse will be generated by the Print Compare circuit which will be routed to the Print Matrix. In this matrix, the pulse will be directed to the hammer drive circuits for the print hammer in the first printing position and firing of the hammer will be initiated.

On the second memory cycle, the fourth position of memory (204) will be addressed by the "B" address register and the character counter will be advanced two steps. This operation continues until each character has been aligned with each print position.

Checking of the print operation in the SPACE unit has not been completely resolved. However, the method of detecting that a hammer has fired by noting the current change in a secondary winding on the hammer magnet yoke will be used. This will set a core in the ninth plane above the print storage area corresponding to the print position under consideration. Still to be resolved is the most effective and practical method of utilizing this signal.

Carriage

The carriage to be used will be a hydraulic unit and identical to that being incorporated with the Chain Printer with the exception that it will operate on a single rather than a two-speed basis in the basic machine.

Two single character registers are associated with the carriage and are used to hold carriage advance information from the time it is determined by the program until the end of the carriage operation. The two registers are: 1) Skip register which will contain "skip-to" information with its output compared with

a paper locating means such as tape signal or an electronic counter in order to stop the carriage at a predetermined location; and 2) Space register which will be capable of storing only 1, 2, or 3 and will, in conjunction with a line space counter, permit the carriage to advance the prescribed number of line spaces.

Data to these registers is obtained from the "A" register on the "I" portion of a carriage instruction cycle as described in the Cycle Control section of this report.

Print Display

One of the servicing and customer aids included in the machine will be the ability to display any single one hundred character block of storage in printed form. Word marks for that block of storage will appear as an "X" above the print position corresponding to storage location in which they appear.

This function is initiated by the setting of a dial switch on the console to the hundred block notation (1 for 101-200, 2 for 201-300, etc.) and depressing the "Print Display" key. The contents of that storage block will then be printed out from that location without being transferred to the print area. This is done so that the information in the print area will not be lost.

Upon depression of the "Print Display" key the "B" address register is set to the 01 position of the block specified. Successive "B" cycles are taken as in print output; however, since only 100 characters are to be read out, the sensing of 00 instead of 20 in the low order digits of the address register will terminate each print scan. During these "B" cycles the data from the specified block is read through the "B" register to the Print Compare and a normal print operation is followed.

BASIC CYCLES OF MACHINE OPERATION

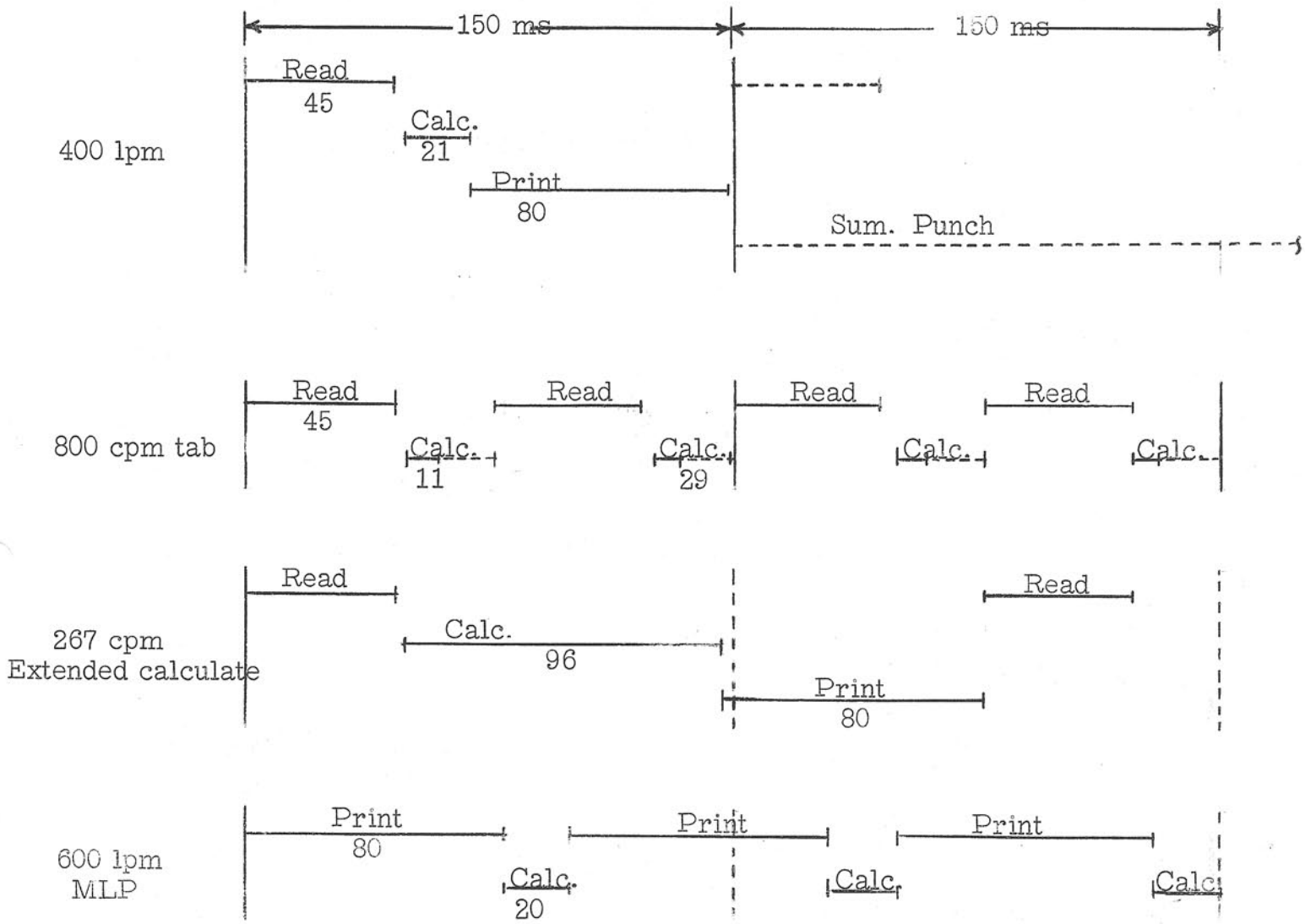
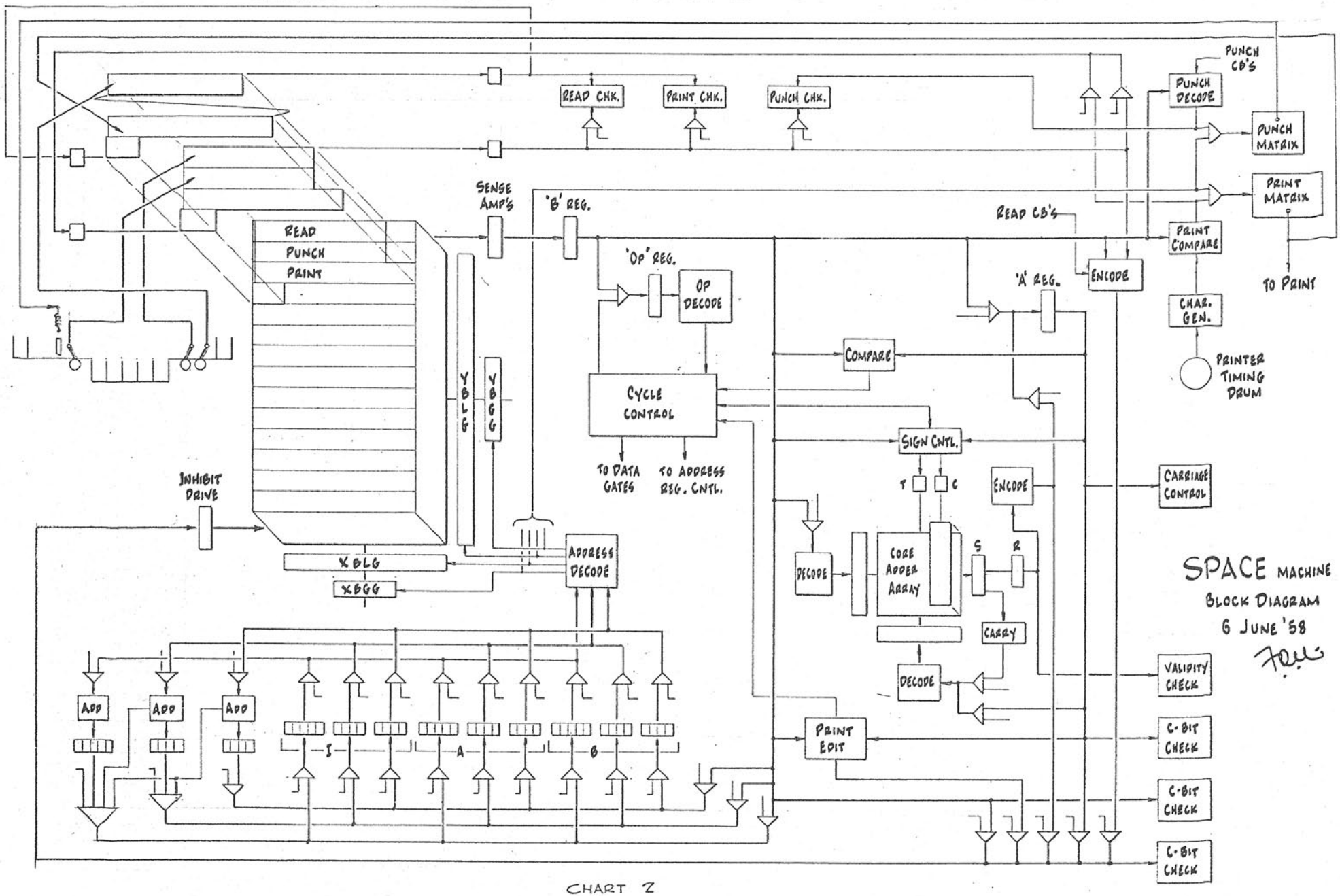


CHART I



SPACE MACHINE
BLOCK DIAGRAM
6 JUNE '58

CHART 2

SPACE

VARIABLE-LENGTH INSTRUCTION FORMAT

OP	A/I	B	C/I	I	D	USE
OP						MACHINE CONTROL (FEED, PRINT, PUNCH)
OP	D					CARRIAGE (SKIP TO D, SPACE D)
OP	I					PROGRAM SKIP TO I (NOTE 1)
OP	I	D				CARRIAGE CONTROL AND PROG. SKIP
OP	A	B				TRANSFER A→B, A+B=B' (NOTE 2)
OP	I	B	D			BRANCH TO I IF B CONTAINS NO D.
OP	A	B	C			A+B=C, AxB=C, C/A=B (NOTE 3)
OP	A	B	C	D		AS ABOVE, WITH SHIFT PER D.
OP	A	B	C	I		AS ABOVE, WITH PROGRAM SKIP.
OP	A	B	C	I	D	AS ABOVE, WITH SKIP AND SHIFT.

NOTES

1- OP. CODE MAY BE A MACHINE CONTROL OP. CODE.

2- ALSO EDITING IN BASIC MACHINE (A [MOD] B = B')

3- ALSO OP-A-B-I FOR FIELD COMPARING
ALSO FOR EDITING IN EXPANDED MACHINE (A [MOD] B = C)

4- AN INSTRUCTION OF THE FORM OP-I_1-B-I_2-D PERMITS BRANCHING TO I_1 IF B CONTAINS NO D, AND A PROGRAM SKIP TO I_2 IF B DOES CONTAIN D.

CHART #3

17 FEB '58 · 4740-701